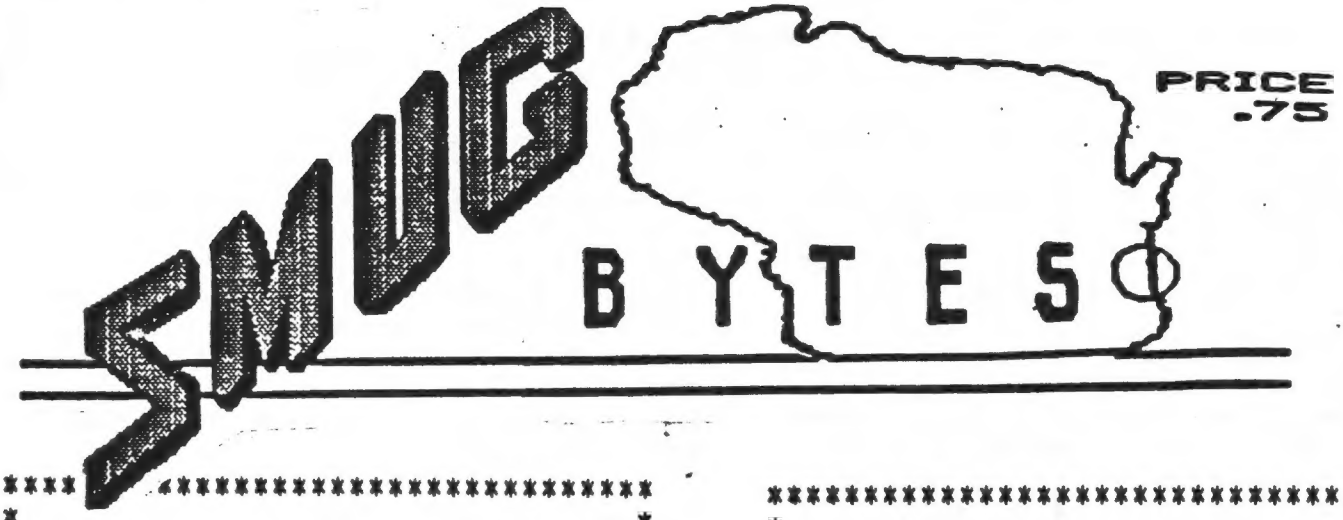



SMUG BYTES
Volume 5, Number 10
October 1988

Subscriptions= $10.00/year
or free with swap. Send to:

SINCLAIR MILWAUKEE USERS GROUP
P.O. Box 101 Butler, WI 53007

---

*THIS MONTH:*

- *Bill On Mscript.*
- *Draw A Holloween Pumpkin*
- *Rudy's "SQ" NOTES*
- *Presidents Messeae*
- *And Other Great Things*

If any articles are copied please credit SMUG Bytes

---

NEXT MEETING DATE: 11/02/88

---

Send all contributions by the last weekend of the month to:

Bill Heberlein
Editor
SMUG BYTES
5052 N. 91st Street.
Milwaukee, WI 53225

## 1988 OFFICERS & WORKING MEMBERS

All Area Codes Are 414

*President - N. Schultz
- 353 4522
*V. Pres. - W. Heberlein
- 527 2191
*Secretary - L. Dreger
- 321 0694
*Treasurer - J. Frohboese
- 377 1385
*Education - R. Cultice
- 542 3591

- Librarians
*Tape/Disk - E. Pawloski
- 1 338 0260
*Book - D. Nickels
- 1 284 9691

* - Sub Group Leaders
*Games - N. Schultz
- 353 4522
*Meeting on 3rd Sat. of the month*
*Hardware - G. Kraemer
- 421 0179
*No set date. Call for info.
*Spectrum - R. Hilsmann
- 251 5291
*Meeting on 2nd Wed. of the month*
*RP/M - R. Cultice
- 542 3591
*Meeting on 4th Wed. of the month*
*QL - R. Hilsmann
- 251 5291
*Meeting date see Spectrum group *

NOTE NEITHER ANY AUTHOR OR SMUG BYTES TAKES ANY RESPONSIBILITY FOR ANY HARDWARE MODIFICATIONS TO YOUR EQUIPMENT.)

## FROM THE PRES.

Well here is our first meeting with the new date and place. I want to start with a Thank You to Ester for getting us this great location.

Please note the following two ham/computer fest's comming up. First there is the LATE FALL HAM FEST at the Lake County Fairgrounds, routes 45 & 120, Grayslake, IL. 7 am till ? Admission $3. Date 10/30/88.

The other one is the 6.91 FRIENDLY FEST. Serb Hall. 51st & Oklahoma, Milwaukee. Admission $3. Date 11/12/88.

Please note Dick has started "C" classes for all those interested in "C" on the QL. The first class was held on Sunday Oct. 2, 1988. For the next date contact Dick Cultice. Also contact Dick if you want any other type of class. He is our Educational Director and he needs the work.

Reminder to all you members dues are coming up for most of you plus the Annual meeting in January.

One more thing please note the change in the Spectrum group meeting date. They will now meet on the second Wednesday of the month instead of the third Wednesday.

## "SQ" NOTES

BY R.A.HILSMANN

Some more ramblings about things not fully covered in the TIMEX MANUAl this month. Looks like the time between News-letters is shrinking, or else I am to busy with other things.

A look at what the Sinclair Manual has to say about Mathematical functions which are also only covered in the Timex Manual in Appendices. Perhaps you feel that this is a waste of time, since most of you have paid attention in High School, and know all about those, but what about the ones of you who did not? Well do not be suprised if you learn something anyways.

Chapter 10 of the Sinclair Manual deals with the mathematics that your computer can handle. It covers the operation ↑ (raising to the power), the functions EXP and LN, and the trigonometrical functions SIN, COS, TAN and their inverses ASN, ASC, and ATN.

↑ and EXP

You can raise one number to the power of another - that means "multiply the first number by itself the second number of times". This is normaly shown by writing the second number just above and to the right of the first number; but obviously this would be difficult to do on a computer so the ↑ symbol was selected instead. For example, the power of 2 are

```
2 ↑ 1 = 2
2 ↑ 2 = 2*2 = 4
2 ↑ 3 = 2*2*2 = 8
2 ↑ 4 = 2*2*2*2 = 16
```

At its most elementary level, "a↑b" means "'a' multiplied by itself 'b' times", but this only makes sence if 'b' is a positive number. To find a definition that works for other values of 'b', consider the rule

```
a ↑ (b+c) = a↑b*a↑c
```

Notice that ↑ has a higher priority than * and / so that when there are several operations in one expression, the ↑s are evaluated before the *s and /s. You should not need much convincing that this works when 'b' and 'c' are both positive whole numbers; but if you decide that you want it to work even when they are not, then you find yourself compelled to accept that

```
a ↑ 0 = 1
a ↑ (-b) = 1/a↑b
```

a ↑ (1/b) = the bth root of 'a', which is the number that you have to multiply by itself 'b' times to get 'a'

and

```
a ↑ (b*c) = (a↑b) ↑ c
```

If you have never seen any of this before then don't try to remember it now: just try to remember that

```
a ↑ (-1) = 1/a
and
a ↑ (1/2) = SQR a
```

maybe when you are familiar with these the rest will begin to make sense.

Experiment with all this by trying this program:

```
10 INPUT a,b,c
20 PRINT a↑(b+c),a↑b*a↑c
30 GO TO 10
```

Of course, if the rule I gave earlier is true, then each time around, the two numbers that the computer prints out will be equal. (Note - because of the way the computer works out ↑, the number on the left - 'a' in this case - must never be negative.)

An example of what this function can be used for is that of compound interest. Suppose you keep some of

your money in a savings account which pays 15% interest per year (what a deal). Then after one year you will have not just the 100% that you had anyway, but also the 15% interest your money has earned, making altogether 115% of what you had originally. To put it another way, you have multiplied your sum of money by 1.15. After another year, and if the sum of money you have put in the account originally has not changed, the same will have happened again, so that you will then have 1.15*1.15=1.15↑2=1.3225 times your original sum of money. In general, after 'y' years, you will have 1.15↑y times what you started out with.

If you try this line

```
FOR y=0 TO 100: PRINT  y,10*1.15↑y: NEXT y
```

you will see that even starting out with just $10, it all adds up quite fast, and what's more, it increases faster and faster as time goes on. (Although, you might still find that it doesn't keep up with inflation.)

This type of behaviour, where after a fixed interval of time some quantity multiplies itself by a fixed proportion, is called exponential growth, and it is calculated by raising a fixed number to the power of the time.

Suppose you did this:

```
10 DEF FN a(x)=a↑x
```

Here, 'a' is more or less fixed by LET statements: its value will correspond to the interest rate, which changes only ever so often.

There is a certain value for 'a' that makes the function FN 'a' look especially pretty to the trained eye of a mathematician: and this value is called 'e'. Your computer has a function called EXP defined by

```
EXP x = e↑x
```

Unfortunately, 'e' itself is not an especially pretty number: it is an infinite non-recurring decimal. You can see its first few decimal places by doing

```
PRINT EXP 1
```

because EXP 1 = e↑1 = e. Of course, this is just an approximation. You can never write down 'e' exactly.

LN

The inverse of an exponential function is a logarithmic function: the logarithm (to base 'a') of a number 'x' is the power to which you have to raise 'a' to get the number 'x', and it is written log a\x, normally shown by writing 'a' just below and to the left of 'x'. Thus by definition 'a↑log a\x=x'; and it is also true that 'log (a↑x)=x'.

You may already know how to use base 10 logarithms for doing multiplications; these are called common logarithms. Your computer has the function LN which calculates logarithms to the base 'e'; these are called natural logarithms. To calculate logarithms to any other base, you must divide the natural logarithm by the natural logarithm of the base:

```
log a\x= LN x/ LN a
```

PI

Given any circle, you can find its perimeter (distance around its edge; called circumference) by multiplying its diameter (width) by a number called PI (Pi is a Greek p, and is used because it stands for the word perimeter).

Like 'e', PI is an infinite non-recurring decimal; it starts off as 3.141592653589...The function PI on your computer returns this number. Try PRINT PI.

SIN, COS & TAN; ASN, ACS & ATN

The trigonometrical functions measure what happens when a point

moves round a circle. Here is a circle of radius 1 (1 what? It does not matter, as long as you keep the same unit all the time. You could also pick any other number!) and a point moving around it. The point startedat the 3 o'clock position, and then moves in a counter clockwise direction. I have also drawn in two lines called axes through the center of the circle. The one from 9 o'clock to 3 o'clock is called the x-axis, and the one from 6 o'clock to 12 o'clock is called the y-axis.

To specify where the point is, or how far it has moved around the circle from its 3 o'clock position; let's call this distance 'a'. It is known that the circumference of the circle is 2*PI (because its radius is 1 and its diameter is thus 2): so when it has moved a quarter way around the circle, a=PI/2; when it has moved half way around, a=PI; and when it has moved all the way, a=2PI.

Given the curved distance around the circle 'a', two other distances you may like to know are how far the point is to the right of the y-axis, and how far it is above the x-axis. These are called, respectively, the cosine and sine of 'a'. The function COS and SIN on your computer will calculate these.

Note that if the point goes to the left of the y-axis, the cosine becomes negative; and if the point goes below the x-axis, the sine also will become negative.

Another property is that once 'a' has reached 2PI, the point is back where it started from and the sine and cosine starts with the same values all over again:

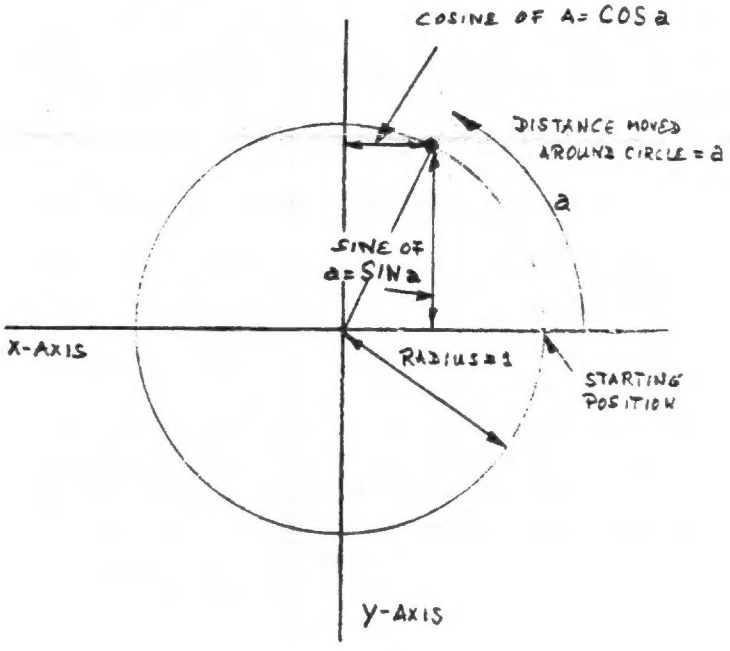
```
SIN (a+2*PI)=SIN a
COS (a+2*PI)=COS a
```

The tangent of 'a' is defined to be the sine divided by the cosine; the corresponding function is called TAN.

Sometimes we have to work these functions out in reverse, finding the value of 'a' that gives the sine, cosine or tangent. The functions to do this are called arcsine (ASN), arccosine (ACS) and arctangent (ATN).

In the diagram of the point moving around the circle, look at the radius joining the center to the point. You should be able to see that the distance called 'a', the distance that the point has moved around the circle, is a way of measuring the angle through which the radius has moved away from the x-axis. When a=PI/2, the angle is 90 degrees; when a=PI the angle is 180 degrees; and around to when a=2PI (a=2*PI), and the angle is 360 degrees (a full circle). You might as well forget about degrees; and measure the angle in terms of 'a' alone: say then that you are measuring the angle in radians. Thus PI/2 radians = 90 degrees and so on.

Remember that on your computer SIN, COS etc. use radians and not degrees. To convert degrees to radians, devide by 180 and multiply by PI; to convert back from radians to degrees, divide by PI and multiply by 180.



Have fun, like always, I hope this has helped to clarify some of the uses for each function.

A SINCLAIR PC?

Just got the new fall catalog from SHARP's. A SINCLAIR MSDOS PC?? You've got to be kidding, $ 699.95 to $ 1449.95, depending on what kind of setup you like!

I don't wish to sound negative, but isn't this like inventing the wheel all over again? What happened to being different or unique? Isn't this why we are still stuck on the 2068 or the QL, because they are unique? Unique in that they are more transparent etc.

Why would I buy something just that ordinary, something I dont have to fight to get software for? Something I can buy at the corner computer store? Everybody has them; IBM clones that is, and probably cheaper! I dont see it, but then that's my opinion. Uncle Clyde must be crunching his teeth thinking about such cliché.

But for all you IBM lovers, DISCOVER is available, $ 49.95 will allow the QL to write or read to any PC disk, and it will also let you view or delete files, but will it let you run PC programs on your QL?? I guess not, just read and write data to or from a PC formated disk it looks like. Or is it the other way around? A mistery.

But yes, there is another one coming, something called an IBM EMULATOR for the QL! Soon they say.

Dont forget the SWAPFEST at the WAUKESHA EXPO on the 30th of OCT. Just thought I mention it since we hadn't talked about it at the last meeting. Another thing, don't forget to renew your supscription to 2068 UPDATE, it's worth it.

Well this should do it for this month, till' next time,

Your #3 RUDY.....

# Bill on Mscript

Well this is a reminder what you see is not necessaraly what you want. Watch out. Instructions may call for a special character and most likly the special character is generated by a combination of MSCRIPT keys not gotten from the key board. For example the ↑. If you see this symbol it probably is the SYMBOL SHIFT, CAPS SHIFT, N and not the SYMBOL SHIFT H keys.

I just got my update of MSCRIPT from Jack Duhaney so next month I hope to have some new information for you and a little review of it.

The following are the setups for page numbering.

>eb=/ SMUG Bytes/$/ August 1988/

sets up a "bottowm of page" line on all even pages.

>ob=/ August 1988/$/ SMUG Bytes/

sets up a "bottom of page" line on all odd pages.

If you use et or ot instead of eb or ob the page numbering will be on the "top of page" not the bottom of the page. Also the $ is not required if you do not want page numbering. There are other commands that go with these top and bottom commands but to understand them best you should play with them. They are not difficult to use so go at it.

Try the MSCRIPT you'll like it.

Bill

**The following is a nice little program. It is just in time for the October holloween season.**

```
   1 REM       PUMPKIN
       © GREG STEINER 1988
  10 BORDER 0: PAPER 6: CLS : IN
K 9
  20 PLOT 127,160: DRAW 0,-155,1
.2*PI
  30 DRAW 0,155,1.2*PI
  40 PLOT 127,160: DRAW 0,-155,.
5*PI
  50 DRAW 0,155,.5*PI: DRAW 0,-1
55,.9*PI: DRAW 0,155,.9*PI
 300 FOR y=175 TO 0 STEP -1
 310 FOR x=255 TO 0 STEP -1
 320 IF POINT (x,y) THEN GO TO _
20
 330 PLOT x,y
 340 NEXT x
 350 NEXT y
 420 FOR y=0 TO 175
 430 FOR x=0 TO 255
 440 IF POINT (x,y) THEN GO TO 1
80
 450 PLOT x,y
 470 NEXT x
 480 NEXT y
 500 REM stem
 510 INK 4
 520 FOR i=124 TO 135
 530 PLOT i,156
 540 DRAW 5,19,-PI
 550 NEXT i
 590 INK 0
 600 REM face
 610 REM eyes
 620 FOR y=100 TO 150
 630 FOR x=-40+y TO 210-y
 640 PLOT x,y
 650 PLOT x+85,y
 660 NEXT x
 670 NEXT y
 700 REM nose
 710 FOR y=70 TO 100
 720 FOR x=y+45 TO 210-y
 730 PLOT x,y
 740 NEXT x
 750 NEXT y
 760 REM mouth
 765 PLOT 64,65
 770 FOR p=.3 TO .56 STEP .02
 780 DRAW 126,0,p*PI
 785 DRAW -126,0,-(p+.01)*PI
 790 NEXT p
 800 LET h$="HH  AA  PL  PL  YO
  W   E   E   N  "
 810 FOR i=4 TO 38 STEP 2
 820 INVERSE 1: PRINT AT i/2-1,0
;h$(i-3);AT i/2-1,31;h$(i-2)
 830 NEXT i
1000 SAVE "PUMPKIN" LINE 1
```

PUMPKIN CK TYPE report:

```
1       :  51  :  4314
10      :  29  :  2178
20      :  53  :  3711
30      :  31  :  2435
40      :  52  :  3234
50      :  91  :  6852
300     :  30  :  2194
310     :  30  :  2270
320     :  19  :  2206
330     :   5  :   847
340     :   3  :   632
350     :   3  :   633
420     :  21  :  1638
430     :  21  :  1714
440     :  19  :  2278
450     :   5  :   847
470     :   3  :   632
480     :   3  :   633
500     :   6  :  1147
510     :   9  :   544
520     :  23  :  1851
530     :  13  :  1104
540     :  21  :  1220
550     :   3  :   617
590     :   9  :   536
600     :   6  :  1078
610     :   5  :  1155
620     :  23  :  1849
630     :  27  :  2263
640     :   5  :   847
650     :  14  :  1343
660     :   3  :   632
670     :   3  :   633
700     :   6  :  1130
710     :  22  :  1691
720     :  26  :  2364
730     :   5  :   847
740     :   3  :   632
750     :   3  :   633
760     :   7  :  1278
765     :  19  :  1116
770     :  32  :  4687
780     :  23  :  1502
785     :  37  :  3079
790     :   3  :   624
800     :  43  :  3567
810     :  28  :  1778
820     :  93  :  4440
830     :   3  :   617
1000    :  19  :  1930
```